USF UNIVERSITY OF SAN FRANCISCO

CHANGE THE WORLD FROM HERE

Regular Epressions

Quick Guide

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science

Definition

There is no official standard for regular expressions!

(So, there is no official definition.)

Colloquially, think of them as a way to find a **text pattern** for **searching, editing,** and/or **replacing text.**

http://docs.oracle.com/javase/8/docs/technotes/guides/collections/index.html

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

RegEx Flavors

The syntax & additional properties supported by the regex engine.

BRE (Basic)	sed
ERE (Extended)	grep
PCRE	Perl, PHP, Tcl, etc..

Java's engine (via Pattern class) is Perl-ish

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

Character Classes

- Matches to one and ONLY one character in a set of characters
- [abc] - a, b, or c
- [^abc] - Any character except a, b, or c
- [A-zA-Z] - a through z, or A through Z

Metacharacters

A **metacharacter** is a character with special meaning in the regex engine. Some metacharacters supported by the Java regex engine are:

- \d - A digit
- \D - A non-digit
- \s - A whitespace
- \S - A non-whitespace
- \w - A word character (letters, numbers, an underscore)
- \W - A non-word character
- . - Any character (line terminators maybe…)
- * - Wild
- \[- Open bracket “[“ character (literal escape)

Quantifiers

Used to specify the number of occurrences

- X? - X, once or not at all
- X* - X, zero or more times
- X+ - X, one or more times
- X{n} - X, exactly *n* times
- X{n,} - X, at least *n* times
- X{n,m} - X, at least *n* but not more than *m* times

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

Some examples

- [Aa] - A or a
- \d+ - A number
- 0[xX](0-9a-f-A-F)+ - A hex number
- [a-fn-s] - A char between a to f OR n to s

Java Classes

The java.util.regex package consists of two primary classes:

- Pattern
- Matcher

PatternSyntaxException is a runtime exception

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

The Pattern Class

- A regular expression (as a string) is compiled into an instance of this class.
- The resulting pattern is used to create a Matcher object that matches character sequences against the regular expression.

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

The Matcher Class

```
Pattern p = Pattern.compile(“a*b”);

Matcher m = p.matcher(“aaaaababaaaab”);


m.matches()       // attempts to match the entire input
                  //  sequence against the pattern.

m.lookingAt()     // attempts to match starting at the
                  //  beginning of the input sequence

m.find()          // scans and looks for the next
                  //  subsequence that matches
```


Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

Output Operations

Once a match has been found, you can use Matcher methods to parse and interact with the results:

start() - returns the start index of the match

end() - returns the offset after the last character matched

group() - returns the input subsequence matched by the previous match

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

Regular Expressions as Strings

```
String regex = “\d”;        // compiler error!!!

String regex = “\\d”;       // this is okay

String regex = “\.”;        // nope :(

String regex = “\\.”        // yep :)
```


Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science
UNIVERSITY OF SAN FRANCISCO

Chris Bryan ◆ cbryan2@usfca.edu
Department of Computer Science